BRL-TR-2961

(2)

DTIC FILE COPY

TECHNICAL REPORT BRL-TR-2961

BRL

1938 – Serving the Army for Fifty Years – 1988

AD-A202 413

THE TIMELINE ANALYSIS MODEL

JOSEPH K. WALD

DECEMBER 1988

U.S. ARMY LABORATORY COMMAND

BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

DESTRUCTION NOTICE

Destroy this report when it is no longer needed. DO NOT return it to the originator.

Additional copies of this report may be obtained from the National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161.

The findings of this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) BRL-TR-2961	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION US Army Ballistic Research Laboratory	6b. OFFICE SYMBOL (If applicable) SLCBR-DD-T
7a. NAME OF MONITORING ORGANIZATION	
6c. ADDRESS (City, State, and ZIP Code) Aberdeen Proving Ground, MD 21005-5066	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
The Timeline Analysis Model

12. PERSONAL AUTHOR(S)
Wald, Joseph K.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 15JUL88 TO 15AUG88	881015	36

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
12	03	
12	04	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
timeline components, Monte Carlo model, statistical distributions, cumulative timeline distribution

19. ABSTRACT (Continue on reverse if necessary and identify by block number)
The engagement sequence, or timeline, of a weapon system consists of a set of functions or processes, some of which must occur sequentially, while others may run in parallel. Each of these processes may be described by a different mathematical or statistical model. In order to be effective, a weapon system may have to complete its entire timeline within a certain time limit. The Timeline Analysis Model is a computer program which combines the models of the various timeline components to produce a cumulative total timeline distribution. From this distribution one can determine the probability that the weapon system will be successful in meeting its time limit requirement.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Joseph K. Wald	22b. TELEPHONE (Include Area Code) 301-278-6669
22c. OFFICE SYMBOL SLCBR-SE-B	

DD Form 1473, JUN 86 Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

The following page is blank.

TABLE OF CONTENTS

The following page is blank.

LIST OF FIGURES

The following page is blank.

THE TIMELINE ANALYSIS MODEL

I. INTRODUCTION

One of the measures by which we evaluate tactical weapon systems is the time it takes them to complete a hostile engagement. Such evaluations are complicated by the fact that battlefield engagements involving modern weapon systems require the performance of many different functions, which are generally described by a variety of mathematical and statistical models. During some parts of the engagement sequence, these functions may occur sequentially, during other parts, they may be concurrent. These factors make more difficult the modeling of engagement timelines.

In this report, we introduce the **Timeline Analysis Model**, a computer program that blends disparate mathematical and statistical models to produce a cumulative total engagement timeline distribution. In section II, we define precisely what we mean by a timeline and give a description of the model. Section III contains detailed instructions on how to use the model, as well as a detailed example of its use. In section IV, we indicate ways in which the model can be extended to investigate timelines of higher orders of complexity. A printout of the computer program can be found in the appendix.

II. MODEL DESCRIPTION

The **Timeline Analysis Model** is a Monte Carlo Simulation that produces a cumulative total engagement timeline distribution from the component parts of the timeline. In particular, each component of the timeline is simulated by one or more mathematical or statistical models, a random number draw is made (in the case of stochastically modeled components) to obtain a specific realization of the time it takes to complete that timeline component, and a single realization of the whole timeline is obtained by adding the component times. This procedure is repeated a large number of times and a cumulative distribution results. An example of such a cumulative distribution is shown in figure 1.

Before giving a precise definition of a timeline, some preliminary terminology is in order. An "element" is defined to be a process extending in time that can be modeled by specifying the parameters of a single elementary mathematical model or statistical distribution. For example, the process of infrared detection can be modeled using the parameters of the exponential distribution. A "string" is defined to be a set of sequential elements. The three elements "trigger pull", "bullet flyout", "kill assessment" form an example of a string. A "cell" is a set of parallel strings. Cells are used to model pieces of the timeline in which several operations are going on simultaneously. For example, we would use a cell to model the situation in which a weapon system has several detection devices operating at the same time. Using the above terminology, we now define a "timeline" to be a set of sequential cells.

Using the schematic timeline depicted in figure 2, we can illustrate the timeline terminology defined above. This sample timeline consists of nine cells with a variable number of strings per cell and a variable number of elements per string. The line segment connecting each pair of dots represents a single element. Cell 6 shows that a cell can consist of a single string, while cell 1 shows that a string (and even a whole cell) can consist of a single element. This example shows that the number of different pathways through a timeline can be quite large; in this case there are $1 \times 6 \times 1 \times 3 \times 2 \times 1 \times 2 \times 2 \times 2 = 288$ ways to traverse the timeline.

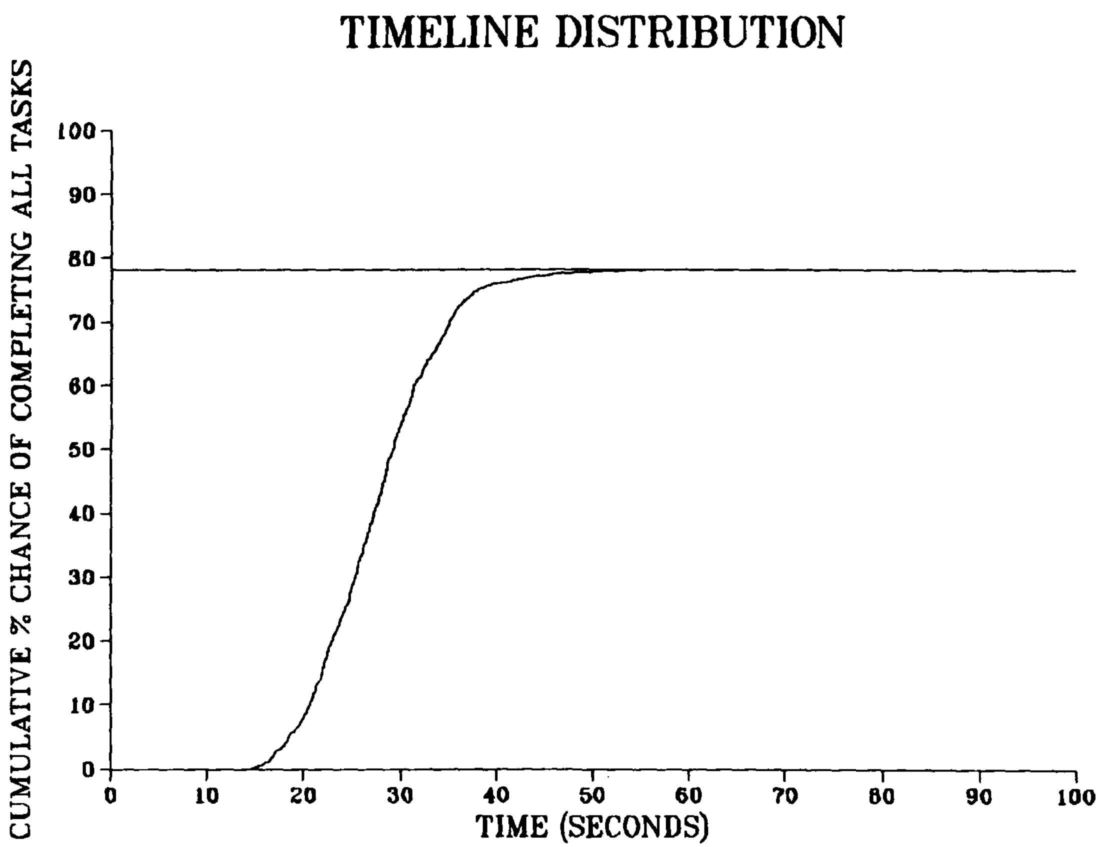

Figure 1. Example of a Cumulative Timeline Distribution.

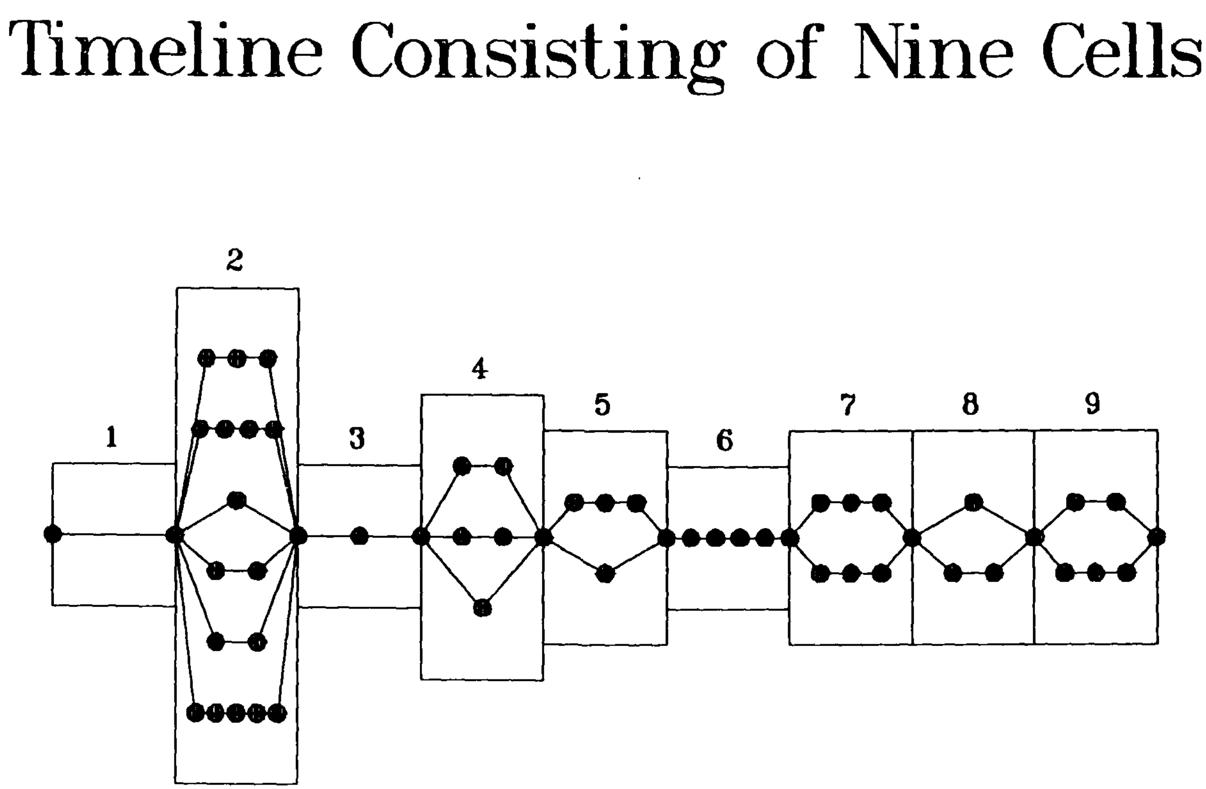

Figure 2. Schematic Timeline "Cell-String-Element" Structure.

When a cell contains two or more strings, we allow two distinct interpretations. Either all of the processes in all of the the strings must be completed before moving on to the next cell, or we move on to the next cell as soon as the processes in any one of the strings are completed. The first case corresponds to assigning to the cell the *maximum* time of all of the component strings, while the second case corresponds to assigning to the cell the *minimum* time of all of the component strings. This process is complicated by the fact that, due to use of the exponential distribution, there are some cases in which some of the processes in a string are never completed. It is even possible that all of the strings in a cell suffer this fate. When this happens, the timeline cannot be completed. The horizontal line at the 78th percentile in figure 1 indicates that this state of affairs occurred in 22 percent of the Monte Carlo replications in that example.

The model input and output will be discussed in detail in the next section. Suffice it to say here that the user must provide both the timeline structure and the relevant parameters and can receive both tabular and graphical output. The graphics associated with the program were produced using the DISSPLA graphics package, built by Integrated Software Systems Corporation. If this package is not available, the user may excise the appropriate portion of the code (found in subroutine OUTPUT) without otherwise affecting the operation of the program. Of course, alternate graphics packages may be substituted by the user. In any case, the tabular output will always be available.

The user must specify the model to be used for each element as well as the corresponding parameters. There are six choices; a constant, a uniform distribution on an interval, a Gaussian distribution, an exponential distribution

$$p_{cum}(t) = p_{\infty}\,[1 - e^{-t/tbar}]$$

(used in electro-optical detection modeling), a lognormal distribution (used in modeling timeline elements for tanks), and a distribution based on the "independence formula"

$$p_{cum} = 1 - (1 - p)^{n}$$

(used in modeling radar detection). There are two parameters corresponding to each distribution, as shown in table 1.

TABLE 1. Model Distribution Parameters.*

distribution	code	first parameter	second parameter
constant	1	value of constant	dummy (not used)
uniform	2	lower endpoint of interval	upper endpoint of interval
Gaussian	3	mean	standard deviation
exponential	4	p-infinity (threshhold probability)	tbar (mean time to detect)
independence	5	probability (of single scan detection)	delta t (scanning period)
lognormal	6	median	standard deviation

The **Timeline Analysis Model** was designed for the purpose of comparing the timeline performance of existing and conceptual weapon systems against a requirement that in a given percentage of opportunities, the system must complete all of the tasks in the timeline within a certain time limit. To this end, the user can specify these requirement parameters and the program will compute the achieved percentage of successful completions by the given time. The requirement can be superimposed on the graphical output.

*WARNING: Improper or careless selection of parameters may cause negative times to be included in the timeline calculation.

III. RUNNING THE PROGRAM

In this section, we give a line by line description of the input subroutine followed by a detailed example of the use of the model.

Except for the third line of input, which is used as both the table heading and the graph title, and is read in under "A50" format, all input is read in under free format. Line 1 contains NUCASE, the number of different timelines to look at during this run.* If NUCASE is greater than 1, then the rest of the input is repeated NUCASE times. In line 2, the output control parameters are read in. These parameters are summarized in table 2.

TABLE 2. Output Control Parameters.

parameter	values	function controlled
IGRAPH	0 (no graphics produced)	graphics output
	1 (graphics produced)	
ITABLE	0 (no table produced)	tabular output
	1 (requirement comparison only)	
	2 (requirement + cumulative distribution)	
IREQ	0 (requirement not displayed)	graphical display of requirement
	1 (requirement displayed)	
TMAX	positive real number (seconds)	maximum time displayed in graphics
NINCR	positive integer	(approximate) number of lines in table

Line 3 contains RUNID, the table heading/graph title. Line 4 contains NUMREP, the number of Monte Carlo replications (typically 1000), and XREQ and YREQ, the time limit (in seconds) and percentage requirements, respectively. Line 5 consists of NCELL, the number of cells in the timeline.

The amount of input remaining is variable, depending on the details of the timeline structure. For each cell, a line is read in which contains NSTRNG(K), the number of strings in the Kth cell, and TYCELL(K), an integer parameter indicating whether the program is to calculate the maximum value (TYCELL(K) = 1) or the minimum value (TYCELL(K) = 2) of the string times for the Kth cell.

The final set of input information is contained in a triply nested set of "DO LOOPs". For each string in each cell, the program reads in NELT(J,K), the number of elements in the Jth string of the Kth cell, and for each of these elements, reads in DISTR(I,J,K), PAR1(I,J,K), PAR2(I,J,K), the choice of distribution and parameters for the Ith element of the Jth string of the Kth cell. The value of DISTR(I,J,K) can be found under the "code" column of table 1.

We now present a detailed example of the use of the model. Let us create a hypothetical weapon system and a hypothetical timeline requirement, develop an input set, and run the model to determine if the requirement is met.

The first cell of our timeline consists of four strings, each containing one element. These four strings correspond to different detection devices which are simultaneously attempting to detect a target. Let us assume that these devices are a radar, an infrared device, a pair of binoculars, and the naked eye. As soon as a target is detected by any of these devices, the detection phase is over and we move on to the second cell. This cell contains two strings. One string has three elements - turret slew, lock-on, and filter settling; the other contains only one - target identification. The processes in both strings must be completed before trigger pull is permitted. The third cell contains just one string with two elements - ordnance flyout and kill assessment. The structure of our timeline is presented in table 3.

*NUCASE is read in at the beginning of the program, but all other input is entered via subroutine INPUT.

TABLE 3. Sample Timeline Structure.						
cell number	cell code	string number	element number	distribution code	parameter 1	parameter 2
1	2	1	1	5	1.00	0.25
		2	1	4	0.70	14.50
		3	1	4	0.40	25.80
		4	1	4	0.10	100.00
2	1	1	1	2	0.00	6.25
			2	6	3.50	1.00
			3	1	3.00	0.00
		2	1	4	0.85	12.00
3	1	1	1	3	5.50	0.50
			2	3	5.00	1.00

Certainly, the timeline could have been more finely partitioned, but each distribution was given at least one chance to be heard from.

Figure 3 contains an input set for our timeline. The output parameters are set so as to give the maximum amount of output, both tabular and graphical. The timeline requirement is such that the system must complete the timeline in no more than 30 seconds, in no less than 75 percent of the Monte Carlo replications. Figures 4 and 5 contain the output corresponding to this input. Unfortunately, our system failed to satisfy its timeline requirement.

IV. EXPANDING THE MODEL

There are two ways to expand the **Timeline Analysis Model**. One way is by providing additional modeling options for the description of the elements; the other is by expanding the complexity of the definition of a timeline.

In order to add another modeling option for the elements, it is merely necessary to append to the program a subroutine containing the computations corresponding to the new distribution, and expand the "IF-THEN-ELSE" blocks in subroutine TCELL where the distribution subroutines are called.

As to the matter of increased complexity, one can easily imagine whole timelines playing the part of "superstrings" in a "supercell", "supercells" becoming "supersuperelements" of a "supersuperstring", and so on *ad infinitum*. This process is conceptually easy; however, the computational "bookkeeping" quickly becomes rather daunting.

In any case, the current version of the model is quite useful in the analysis of a variety of existing and conceptual weapon systems.

```
1
1,2,1,60.,15
SAMPLE RUN
1000,30.,75.
 3
   4,2
   2,1
   1,1
      1
        5 1.00 0.25
      1
        4 0.70 14.50
      1
        4 0.40 25.80
      1
        4 0.10 100.10
      3
        2 0.00 6.25
        6 3.50 1.00
        1 3.00 0.00
      1
        4 0.85 12.00
      2
        3 5.50 0.50
        3 5.00 1.00
```

Figure 3. Sample Input List

```
SAMPLE RUN
SPECIFIED TIME =  30.0    REQUIRED % =  75.0    ACHIEVED % =  63.9
```

%	TIME
0.10	12.15
5.70	17.22
11.40	18.55
17.10	19.48
22.80	20.10
28.50	21.14
34.20	22.10
39.90	23.12
45.60	24.34
51.30	25.66
57.00	27.64
62.70	29.73
68.40	32.91
74.10	37.43
79.80	43.97
85.50	64.16
86.80	150.66

Figure 4. Sample Tabular Output

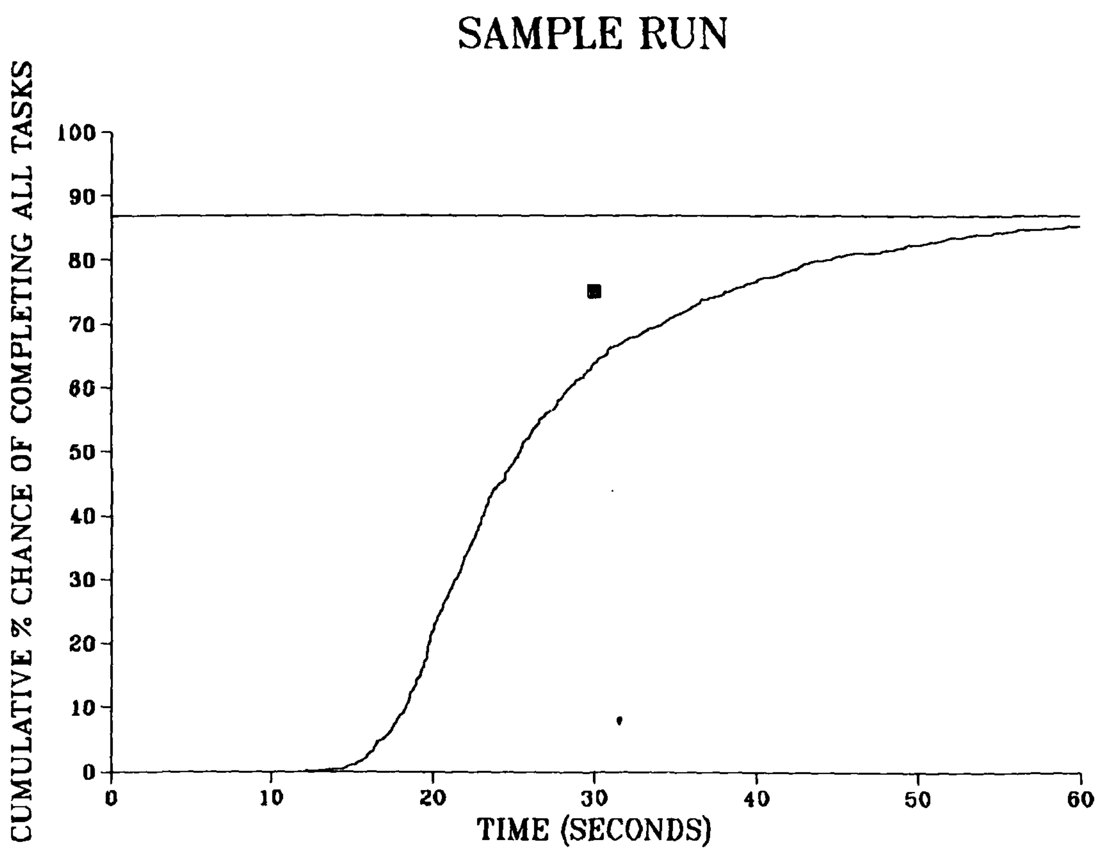

Figure 5. Sample Graphical Output.

APPENDIX

THE TIMELINE ANALYSIS MODEL COMPUTER PROGRAM

```
      PROGRAM TIMES
C
C             THIS PROGRAM DEVELOPES A 'WHOLE
C             TIMELINE' DISTRIBUTION FROM THE
C             DISTRIBUTIONS OF THE TIMELINE
C             COMPONENTS
C
        READ (5,*) NUCASE
        DO 1000 I = 1,NUCASE
          CALL INPUT
          CALL INIRUN
          CALL MONTE
          CALL PREOUT
          CALL OUTPUT
 1000   CONTINUE
C
      END
```

```
C*********************************************************************
      SUBROUTINE INPUT
C*********************************************************************
C
C                 ALL TIMELINE PARAMETERS ARE READ IN
C
      PARAMETER          (MAXCEL = 20
     A                   ,MAXSTR = 5
     B                   ,MAXELT = 10)
C
      COMMON /CONTRL/ IGRAPH,ITABLE,IREQ,TMAX,NINCR
C
      COMMON /MCARLO/ NUMREP,NWIN
C
      COMMON /TLINES/ NCELL,NSTRNG,NELT,DISTR,PAR1,PAR2,XREQ,YREQ
     A                   ,TYCELL
C
      INTEGER             NSTRNG(MAXCEL)
     A                   ,NELT(MAXSTR,MAXCEL)
     B                   ,DISTR(MAXELT,MAXSTR,MAXCEL)
     C                   ,TYCELL(MAXCEL)
C
      REAL                PAR1(MAXELT,MAXSTR,MAXCEL)
     A                   ,PAR2(MAXELT,MAXSTR,MAXCEL)
C
      COMMON /MESAGE/ RUNID
C
      CHARACTER           RUNID*50
C
        READ (5,*) IGRAPH,ITABLE,IREQ,TMAX,NINCR
        READ (5,'(A50)') RUNID
        READ (5,*) NUMREP,XREQ,YREQ
        READ (5,*) NCELL
        DO 1000 I = 1,NCELL
          READ (5,*) NSTRNG(I),TYCELL(I)
 1000   CONTINUE
        DO 4000 K = 1,NCELL
          DO 3000 J = 1,NSTRNG(K)
            READ (5,*) NELT(J,K)
            DO 2000 I = 1,NELT(J,K)
              READ (5,*) DISTR(I,J,K),PAR1(I,J,K),PAR2(I,J,K)
 2000       CONTINUE
 3000     CONTINUE
 4000   CONTINUE
C
        RETURN
      END
```

```
C***********************************************************************
      SUBROUTINE INIRUN
C***********************************************************************
C
C               INITIALIZES 'SEED' FOR UNIFORM
C               RANDOM NUMBER GENERATOR
C
      COMMON /DDURAN/ N1,N2,N3,N4
C
        N1 = 477
        N2 = 510
        N3 = 309
        N4 = 343
C
        RETURN
      END
```

```
C***********************************************************************
      SUBROUTINE MONTE
C***********************************************************************
C
C              THE MONTE CARLO LOOP : THE WHOLE
C              TIMELINE IS COMPUTED 'NUMREP' TIMES
C
      COMMON /MCARLO/ NUMREP,NWIN
C
      COMMON /FAILUR/ INFINY,STOPIT
C
      LOGICAL          INFINY
     A                ,STOPIT
C
        NWIN = 0
        DO 1000 I = 1,NUMREP
          CALL TLINE(T)
          IF  (.NOT. STOPIT)  THEN
            NWIN = NWIN + 1
            CALL TSORT(T)
          ENDIF
 1000   CONTINUE
C
        RETURN
      END
```

```
C**********************************************************************
      SUBROUTINE PREOUT
C**********************************************************************
C
C                 THE RESULTS ARE 'PREPROCESSED' INTO
C                 A FORM SUITABLE FOR OUTPUT
C
      PARAMETER        (MAXREP = 5000
     A                 ,MAXCEL = 20
     B                 ,MAXSTR = 5
     C                 ,MAXELT = 10)
C
      COMMON /CUMDST/ TDCUM,PERCEN,XREQ1,YREQ1
C
      REAL              TDCUM(MAXREP)
     A                 ,PERCEN(MAXREP)
C
      COMMON /MCARLO/ NUMREP,NWIN
C
      COMMON /TLINES/ NCELL,NSTRNG,NELT,DISTR,PAR1,PAR2,XREQ,YREQ
     A                 ,TYCELL
C
      INTEGER           NSTRNG(MAXCEL)
     A                 ,NELT(MAXSTR,MAXCEL)
     B                 ,DISTR(MAXELT,MAXSTR,MAXCEL)
     C                 ,TYCELL(MAXCEL)
C
      REAL              PAR1(MAXELT,MAXSTR,MAXCEL)
     A                 ,PAR2(MAXELT,MAXSTR,MAXCEL)
C
      LOGICAL           NEXT
C
        DO 1000 I = 1,NWIN
          PERCEN(I) = FLOAT(100 * I) / FLOAT(NUMREP)
 1000   CONTINUE
C
        IF      (XREQ .LT. TDCUM(1))  THEN
          YREQ1 = 0.0
        ELSE IF  (XREQ .GE. TDCUM(NWIN))  THEN
          YREQ1 = PERCEN(NWIN)
        ELSE
          NEXT = .TRUE.
          J = 0
 2000     CONTINUE
            J = J + 1
            IF  (XREQ .GE. TDCUM(J) .AND. XREQ .LT. TDCUM(J + 1))  THEN
              FRAC = (XREQ - TDCUM(J)) / (TDCUM(J + 1) - TDCUM(J))
              YREQ1 = (1.0 - FRAC) * PERCEN(J) + FRAC * PERCEN(J + 1)
              NEXT = .FALSE.
            ENDIF
          IF  (NEXT)  GO TO 2000
        ENDIF
C
        RETURN
      END
```

```
C**********************************************************************
      SUBROUTINE OUTPUT
C**********************************************************************
C
C                  THE USER CONTROLS THE TYPE OF OUTPUT
C                  BY SETTING THE VALUES OF THE PARAMETERS
C                  'ITABLE' (FOR TABULAR OUTPUT), AND
C                  'IGRAPH' AND 'IREQ' (FOR GRAPHIC OUTPUT)
C
      PARAMETER         (MAXREP = 5000
     A                  ,MAXCEL = 20
     B                  ,MAXSTR = 5
     C                  ,MAXELT = 10)
C
      COMMON /CONTRL/ IGRAPH,ITABLE,IREQ,TMAX,NINCR
C
      COMMON /CUMDST/ TDCUM,PERCEN,XREQ1,YREQ1
C
      REAL              TDCUM(MAXREP)
     A                  ,PERCEN(MAXREP)
C
      COMMON /MCARLO/ NUMREP,NWIN
C
      COMMON /TLINES/ NCELL,NSTRNG,NELT,DISTR,PAR1,PAR2,XREQ,YREQ
     A                  ,TYCELL
C
      INTEGER           NSTRNG(MAXCEL)
     A                  ,NELT(MAXSTR,MAXCEL)
     B                  ,DISTR(MAXELT,MAXSTR,MAXCEL)
     C                  ,TYCELL(MAXCEL)
C
      REAL              PAR1(MAXELT,MAXSTR,MAXCEL)
     A                  ,PAR2(MAXELT,MAXSTR,MAXCEL)
C
      COMMON /MESAGE/ RUNID
C
      CHARACTER         RUNID*50
C
      REAL              XINF(2)
     A                  ,YINF(2)
C
      REAL              XX(9)
     A                  ,YY(9)
C
      CHARACTER         TITLE*100
C
      DATA  (XX(I),I = 1,9) / 0.,0.,.25,9.25,9.5,9.5,9.25,.25,0. /
      DATA  (YY(I),I = 1,9) / .25,7.25,7.5,7.5,7.25,.25,0.,0.,.25 /
C
        WRITE (11,'(A50)') RUNID
C
        IF  (ITABLE .GE. 1)  THEN
          WRITE (11,'("SPECIFIED TIME = ",F5.1,"    REQUIRED % = "
     A            ,F5.1,"    ACHIEVED % = ",F5.1)') XREQ,YREQ,YREQ1
          IF  (ITABLE .GE. 2)  THEN
            NSTEP = IFIX(FLOAT(NWIN) / FLOAT(NINCR))
            IF  (NSTEP .GT. 0)  THEN
              WRITE (11,'(//,9X,"%",6X,"TIME",/)')
              IF  (NSTEP .GT. 1)  THEN
                WRITE (11,'(F10.2,F10.2)') PERCEN(1),TDCUM(1)
              ENDIF
              DO 1000 I = NSTEP,NWIN,NSTEP
                WRITE (11,'(F10.2,F10.2)') PERCEN(I),TDCUM(I)
 1000         CONTINUE
```

```
              IF  (MOD(NWIN,NSTEP) .NE. 0)  THEN
                WRITE (11,'(F10.2,F10.2)') PERCEN(NWIN),TDCUM(NWIN)
              ENDIF
            ENDIF
          ENDIF
        ENDIF
C
        IF  (IGRAPH .GE. 1)  THEN
          XINF(1) = 0.
          YINF(1) = PERCEN(NWIN)
          XINF(2) = TMAX
          YINF(2) = PERCEN(NWIN)
C
          CALL TK4010(960,0)
          CALL SETDEV(8,8)
          CALL NOBRDR
          CALL PAGE(11.0,8.5)
          CALL PHYSOR(2.0,2.0)
          CALL AREA2D(7.0,4.5)
          CALL BASALF('STANDARD')
          CALL MIXALF('INSTRUCTION')
          CALL TRIPLX
          CALL GRACE(0.)
          CALL YAXANG(0.)
          CALL INTAXS
          CALL XNAME('TIME  (()SECONDS())$',100)
          CALL YNAME('CUMULATIVE % CHANCE OF COMPLETING ALL TASKS$'
     A                ,100)
          CALL ADD(RUNID)
          CALL HEADIN(RUNID,100,1.5,1)
          CALL GRAF(0.,10.,TMAX,0.,10.,100.)
          CALL CURVE(TDCUM,PERCEN,NWIN,0)
          CALL CURVE(XINF,YINF,2,0)
          IF  (IREQ .EQ. 1)  THEN
            CALL MARKER(18)
            CALL CURVE(XREQ,YREQ,1,-1)
          ENDIF
          CALL ENDGR(0)
          CALL ENDPL(0)
          CALL DONEPL
        ENDIF
C
        RETURN
      END
```

```
C***********************************************************************
      SUBROUTINE TLINE(T)
C***********************************************************************
C
C              THIS ROUTINE COMPUTES ONE REALIZATION
C              OF A COMPLETE TIMELINE BY ADDING TOGETHER
C              THE RANDOM TIMES CORRESPONDING
C              TO EACH CELL
C
      PARAMETER       (MAXCEL = 20
     A                ,MAXSTR = 5
     B                ,MAXELT = 10)
C
      COMMON /TLINES/ NCELL,NSTRNG,NELT,DISTR,PAR1,PAR2,XREQ,YREQ
     A                ,TYCELL
C
      INTEGER          NSTRNG(MAXCEL)
     A                ,NELT(MAXSTR,MAXCEL)
     B                ,DISTR(MAXELT,MAXSTR,MAXCEL)
     C                ,TYCELL(MAXCEL)
C
      REAL             PAR1(MAXELT,MAXSTR,MAXCEL)
     A                ,PAR2(MAXELT,MAXSTR,MAXCEL)
C
      COMMON /FAILUR/ INFINY,STOPIT
C
      LOGICAL          INFINY
     A                ,STOPIT
C
        T = 0.
        K = 0
 1000   CONTINUE
          K = K + 1
          CALL TCELL(K,TIME)
          T = T + TIME
        IF  (.NOT. STOPIT .AND. K .LT. NCELL)  GO TO 1000
C
        RETURN
      END
```

```
C***********************************************************************
      SUBROUTINE TCELL(K,TIME)
C***********************************************************************
C
C                THIS ROUTINE COMPUTES ONE REALIZATION
C                OF A COMPLETE CELL BY ADDING TOGETHER
C                THE RANDOM TIMES CORRESPONDING TO THE
C                CONSTITUENT DISTRIBUTIONS OF EACH
C                STRING OF THE CELL AND SELECTING THE
C                LARGEST OF THESE 'STRING TIMES'
C
      PARAMETER        (MAXCEL = 20
     A                 ,MAXSTR = 5
     B                 ,MAXELT = 10)
C
      COMMON /TLINES/ NCELL,NSTRNG,NELT,DISTR,PAR1,PAR2,XREQ,YREQ
     A                 ,TYCELL
C
      INTEGER           NSTRNG(MAXCEL)
     A                 ,NELT(MAXSTR,MAXCEL)
     B                 ,DISTR(MAXELT,MAXSTR,MAXCEL)
     C                 ,TYCELL(MAXCEL)
C
      REAL              PAR1(MAXELT,MAXSTR,MAXCEL)
     A                 ,PAR2(MAXELT,MAXSTR,MAXCEL)
C
      COMMON /FAILUR/ INFINY,STOPIT
C
      LOGICAL           INFINY
     A                 ,STOPIT
C
      REAL              T(MAXSTR)
C
      LOGICAL           COUNTJ(MAXSTR)
C
        IF       (TYCELL(K) .EQ. 1)  THEN
          STOPIT = .FALSE.
          J = 0
 1000     CONTINUE
            J = J + 1
            T(J) = 0.
            I = 0
 2000       CONTINUE
              I = I + 1
              INFINY = .FALSE.
              IF       (DISTR(I,J,K) .EQ. 1)  THEN
                T1 = PAR1(I,J,K)
              ELSE IF  (DISTR(I,J,K) .EQ. 2)  THEN
                CALL URAN(PAR1(I,J,K),PAR2(I,J,K),T1)
              ELSE IF  (DISTR(I,J,K) .EQ. 3)  THEN
                CALL GAUSS(PAR1(I,J,K),PAR2(I,J,K),T1)
              ELSE IF  (DISTR(I,J,K) .EQ. 4)  THEN
                CALL EXPON(PAR1(I,J,K),PAR2(I,J,K),T1)
              ELSE IF  (DISTR(I,J,K) .EQ. 5)  THEN
                CALL INDEP(PAR1(I,J,K),PAR2(I,J,K),T1)
              ELSE IF  (DISTR(I,J,K) .EQ. 6)  THEN
                CALL LOGNOR(PAR1(I,J,K),PAR2(I,J,K),T1)
              ENDIF
              IF  (INFINY)  THEN
                STOPIT = .TRUE.
              ENDIF
              T(J) = T(J) + T1
            IF  (.NOT. STOPIT .AND. I .LT. NELT(J,K))  GO TO 2000
          IF  (.NOT. STOPIT .AND. J .LT. NSTRNG(K))  GO TO 1000
```

```
C
          TIME = 0.
          IF  (.NOT. STOPIT)  THEN
            DO 5000 J = 1,NSTRNG(K)
              TIME = AMAX1(TIME,T(J))
 5000       CONTINUE
          ENDIF
        ELSE IF  (TYCELL(K) .EQ. 2)  THEN
          STOPIT = .FALSE.
          DO 7000 J = 1,NSTRNG(K)
            COUNTJ(J) = .TRUE.
            T(J) = 0.
            DO 6000 I = 1,NELT(J,K)
              INFINY = .FALSE.
              IF        (DISTR(I,J,K) .EQ. 1)  THEN
                T1 = PAR1(I,J,K)
              ELSE IF  (DISTR(I,J,K) .EQ. 2)  THEN
                CALL URAN(PAR1(I,J,K),PAR2(I,J,K),T1)
              ELSE IF  (DISTR(I,J,K) .EQ. 3)  THEN
                CALL GAUSS(PAR1(I,J,K),PAR2(I,J,K),T1)
              ELSE IF  (DISTR(I,J,K) .EQ. 4)  THEN
                CALL EXPON(PAR1(I,J,K),PAR2(I,J,K),T1)
              ELSE IF  (DISTR(I,J,K) .EQ. 5)  THEN
                CALL INDEP(PAR1(I,J,K),PAR2(I,J,K),T1)
              ELSE IF  (DISTR(I,J,K) .EQ. 6)  THEN
                CALL LOGNOR(PAR1(I,J,K),PAR2(I,J,K),T1)
              ENDIF
              IF  (INFINY)  THEN
                COUNTJ(J) = .FALSE.
              ENDIF
              T(J) = T(J) + T1
 6000       CONTINUE
 7000     CONTINUE
          JCOUNT = 0
          DO 7500 J = 1,NSTRNG(K)
            IF  (COUNTJ(J))  THEN
              JCOUNT = JCOUNT + 1
            ENDIF
 7500     CONTINUE
          IF  (JCOUNT .EQ. 0)  THEN
            STOPIT = .TRUE.
          ENDIF
          IF  (.NOT. STOPIT)  THEN
            TIME = 999999999.
            DO 8000 J = 1,NSTRNG(K)
              IF  (COUNTJ(J))  THEN
                TIME = AMIN1(TIME,T(J))
              ENDIF
 8000       CONTINUE
          ENDIF
        ENDIF
C
        RETURN
      END
```

```
C**********************************************************************
      SUBROUTINE TSORT(T)
C**********************************************************************
C
C                   THIS ROUTINE UPDATES THE CUMULATIVE DISTRBUTION
C
      PARAMETER      (MAXREP = 5000)
C
      COMMON /MCARLO/ NUMREP,NWIN
C
C
      COMMON /CUMDST/ TDCUM,PERCEN,XREQ1,YREQ1
C
      REAL                TDCUM(MAXREP)
     A                    ,PERCEN(MAXREP)
C
      LOGICAL             NEXT
C
        IF  (NWIN .EQ. 1)  THEN
          TDCUM(NWIN) = T
        ELSE
          NEXT = .TRUE.
          J = 0
 1000     CONTINUE
            J = J + 1
            IF  (T .LE. TDCUM(J))  THEN
              DO 2000 K = NWIN,J + 1,-1
                TDCUM(K) = TDCUM(K - 1)
 2000         CONTINUE
              TDCUM(J) = T
              NEXT = .FALSE.
            ENDIF
          IF  (J .LT. NWIN - 1 .AND. NEXT)  GO TO 1000
          IF  (J .EQ. NWIN - 1 .AND. NEXT)  THEN
            TDCUM(NWIN) = T
          ENDIF
        ENDIF
C
        RETURN
      END
```

```
C*********************************************************************
      SUBROUTINE URAN(A,B,R)
C*********************************************************************
C
C               MACHINE-INDEPENDENT,UNIFORM (A,B),
C               RANDOM-NUMBER GENERATOR
C
      COMMON /DDURAN/ N1,N2,N3,N4
C
      DATA  M1      / 477 /
     A     ,M2      / 510 /
     B     ,M3      / 309 /
     C     ,M4      / 343 /
C
        K = N4 * M4
        KD = K / (2 ** 9)
        NC1 = K - KD * (2 ** 9)
        K = N4 * M3 + N3 * M4 + KD
        KD = K / (2 ** 9)
        NC2 = K - KD * (2 ** 9)
        K = N4 * M2 + N3 * M3 + N2 * M4 + KD
        KD = K / (2 ** 9)
        NC3 = K - KD * (2 ** 9)
        K = N4 * M1 + N3 * M2 + N2 * M3 + N1 * M4 + KD
        KD = K / (2 ** 9)
        NC4 = K - KD * (2 ** 9)
        N1 = NC4
        N2 = NC3
        N3 = NC2
        N4 = NC1
        R1 = FLOAT(N1) * (2. ** ( -9))
     A     + FLOAT(N2) * (2. ** (-18))
     B     + FLOAT(N3) * (2. ** (-27))
     C     + FLOAT(N4) * (2. ** (-36))
C
        R = A + R1 * (B - A)
C
        RETURN
      END
```

```
C***********************************************************************
      SUBROUTINE GAUSS(AVE,DEV,X)
C***********************************************************************
C
C             DRAWS A RANDOM NUMBER FROM A
C             NORMAL DISTRIBUTION
C
      DATA  A0      / 2.515517 /
     A           ,A1      / 0.802853 /
     B           ,A2      / 0.010328 /
     C           ,B1      / 1.432788 /
     D           ,B2      / 0.189269 /
     E           ,B3      / 0.001308 /
C
      F(X) = A0 + (A1 + A2 * X) * X
      G(X) = 1. + (B1 + (B2 + B3 * X) * X) * X
      H(X) = X - F(X) / G(X)
      W(X) = SQRT(ALOG(1. / X ** 2))
C
        CALL URAN(0.,1.,R)
        IF  (R .LE. .5)  GO TO 1000
          Y = W(1. - R)
          Z = H(Y)
          GO TO 1010
 1000   CONTINUE
          Y = W(R)
          Z = - H(Y)
 1010   CONTINUE
        X = AVE + Z * DEV
C
        RETURN
      END
```

```
C***********************************************************************
      SUBROUTINE EXPON(PINF,TBAR,T1)
C***********************************************************************
C
C               DRAWS A RANDOM NUMBER FROM AN
C               EXPONENTIAL DISTRIBUTION
C
      COMMON /FAILUR/ INFINY,STOPIT
C
      LOGICAL          INFINY
     A                ,STOPIT
C
        CALL URAN(0.,1.,R)
        IF  (R .LE. PINF)  THEN
          T1 = - TBAR * ALOG(1.0 - R / PINF)
        ELSE
          INFINY = .TRUE.
        ENDIF
C
        RETURN
      END
```

```
C***********************************************************************
      SUBROUTINE INDEP(P,DELTAT,T1)
C***********************************************************************
C
C               COMPUTES A RANDOM TIME FROM A DISTRIBUTION
C               BASED ON THE 'INDEPENDENCE' FORMULA
C
      COMMON /FAILUR/ INFINY,STOPIT
C
      LOGICAL          INFINY
     A                ,STOPIT
C
        IF  (P .GT. 0.0)  THEN
          IF  (P .LT. 1.0)  THEN
            CALL URAN(0.,1.,R)
            NSCANS = IFIX(0.5 + ALOG(1.0 - R) / ALOG(1.0 - P))
          ELSE
            NSCANS = 0
          ENDIF
          T1 = DELTAT * FLOAT(NSCANS)
        ELSE
          INFINY = .TRUE.
        ENDIF
C
        RETURN
      END
```

```
C*********************************************************************
      SUBROUTINE LOGNOR(X,S,T1)
C*********************************************************************
C
C              DRAWS A RANDOM NUMBER FROM A
C              LOGNORMAL DISTRIBUTION
C
        CALL GAUSS(0.0,S,R)
        T1 = X * EXP(R)
C
        RETURN
      END
```

```
C*********************************************************************
      SUBROUTINE ADD(X)
C*********************************************************************
C
C             ADD A DOLLAR SIGN AFTER THE LAST NON-BLANK CHARACTER
C
      CHARACTER        X*(*)
C
        I = MIN0(LONG(X) + 1,LEN(X))
        X(I:I) = '$'
        RETURN
C
      END
```

```
C***********************************************************************
C***********************************************************************
      FUNCTION LONG(X)
C***********************************************************************
C***********************************************************************
C               SEARCH FOR LAST NON-BLANK, NON-DOLLAR-SIGN CHARACTER IN X
C
      CHARACTER        X*(*)
C
        I = LEN(X)
 1000   CONTINUE
        IF  (X(I:I) .EQ. ' '  .AND.  I .GT. 1)  THEN
          I = I - 1
          GO TO 1000
        ENDIF
        IF  (X(I:I) .EQ. '$')  THEN
          I = I - 1
        ENDIF
        LONG = I
        RETURN
C
      END
```

The following page is blank.

DISTRIBUTION LIST

No. of Copies	Organization
2	Administrator Defense Technical Info Center ATTN: DTIC-FDAC Cameron Station, Bldg. 5 Alexandria, VA 22304-6145
2	HQDA SARD-TR DAMO-FDD (LTC Milne) Washington, DC 20310
1	Commander US Army Materiel Command ATTN: AMCDRA-ST 5001 Eisenhower Avenue Alexandria, VA 22333-0001
2	Commander Armament R&D Center US Army AMCCOM ATTN: SMCAR-TDC/SMCAR-MSI Picatinny Arsenal, NJ 07806-5000
2	Commander US Army Armament Research Development & Engineering Command ATTN: SMCAR-FSF-B (G. DelCoco)/ SMCAR-CCH-W (A. Warnasch) Picatinny Arsenal, NJ 07801-5000
1	Commander US Army Armament, Research, Development & Engineering Command ATTN: AMCPEO-AR (A. Calabrese) Picatinny Arsenal, NJ 07801-5000
1	Director Benet Weapons Laboratory Armament R&D Center US Army AMCCOM ATTN: SMCAR-LCB-TL Watervliet, NY 12189-4050
1	Commander US Army Armament, Munitions & Chemical Command ATTN: SMCAR-ESP-L Rock Island, IL 61299-6000
1	Commander US Army Aviation Systems Command ATTN: AMSAV-DACL 4300 Goodfellow Boulevard St. Louis, MO 63120-1798
1	Director US Army Aviation Research and Technology Activity Ames Research Center Moffett Field, CA 94035-1099
1	Commander US Army Communications- Electronics Command ATTN: AMSEL-ED Fort Monmouth, NJ 07703
1	Commander US Army Missile Command ATTN: AMSMI-AS Redstone Arsenal, AL 35898-5000
3	Commander US Army Tank-Automotive Command ATTN: AMSTA-DI AMSTA-ZEC, Mr. Hoeltzel (2cys) Warren, MI 48090
1	Commander US Army Tank-Automotive Command ATTN: AMCPM-ABMS (COL J. Longhouser) Warren, MI 48397-5000
1	Director US Army TRADOC Analysis Command ATTN: ATAA-SL White Sands Missile Range, NM 88002-5502
1	Air Force Armament Laboratory ATTN: AFATL/DLODL Eglin AFB, FL 32542-5000
1	Commandant US Army Infantry School ATTN: ATSH-CD-CSO-OR Fort Benning, GA 31905-5400

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander US Army Development and Employment Agency ATTN: MODE-ORO Fort Lewis, WA 98433
1	AFWL/SUL Kirtland AFB, NM 87117
10	Central Intelligence Agency OIR/DB/Standard Room GE47 HQ Washington, DC 20505
1	Project Manager Tank Main Armament Systems ATTN: AMCPM-TMA (COL Mullin) Picatinny Arsenal, NJ 07806-5000
1	Commander US Army Laboratory Command ATTN: AMSLC-TD (R. Vitali) 2800 Powder Mill Road Adelphi, MD 20783-1145
1	Commandant US Army Armor School ATTN: ATSB-CD-ML (LTC(P) L.G. Smith) Fort Knox, KY 40121-5000
2	Commander US Army Training & Doctrine Command ATTN: ATCD-M (COL King/M. Pastel) Fort Monroe, VA 23651-5000
1	Office of the Secretary of the Army Assistant Director for Systems & Technology Assessment ATTN: SARD-TRS (Dr. C. Church) Washington, DC 20310
1	The Rand Corporation P.O. Box 2138 Santa Monica, CA 90401-2138

Aberdeen Proving Ground, MD

Cdr, USATECOM
ATTN: AMSTE-TO-F

Dir, USAMSAA
ATTN: AMXSY-D (K. Myers)
AMXSY-G (J. Kramar)
AMXSY-GA (W. Brooks)
AMXSY-A (W. Clifford)
AMXSY-AD (John Meredith)
AMXSY-C (H. Burke)
AMXSY-MP (H. Cohen)

Cdr, CRDC, AMCCOM
ATTN: SMCCR-RSP-A
SMCCR-MU
SMCCR-SPS-IL

USER EVALUATION SHEET/CHANGE OF ADDRESS

This laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers below will aid us in our efforts.

1. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) ______________________________

2. How, specifically, is the report being used? (Information source, design data, procedure, source of ideas, etc.) ______________________________

3. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided, or efficiencies achieved, etc? If so, please elaborate. ______________________________

4. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) ______________________________

BRL Report Number ________________ Division Symbol ________________

Check here if desire to be removed from distribution list. ____

Check here for address change. ____

Current address: Organization ______________________
Address ______________________

------------------------------FOLD AND TAPE CLOSED------------------------------

Director
U.S. Army Ballistic Research Laboratory
ATTN: SLCBR-DD-T(NEI)
Aberdeen Proving Ground, MD 21005-5066

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE $300

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY LABEL
FIRST CLASS PERMIT NO. 12062 WASHINGTON D.C.

POSTAGE WILL BE PAID BY DEPARTMENT OF THE ARMY

Director
U.S. Army Ballistic Research Laboratory
ATTN: SLCBR-DD-T(NEI)
Aberdeen Proving Ground, MD 21005-9989